

ODM Tools Python: Open Source Software for Managing Hydrologic and Water Quality Time Series Data

Jeffery S. Horsburgh

Stephanie Reeder

James Patton

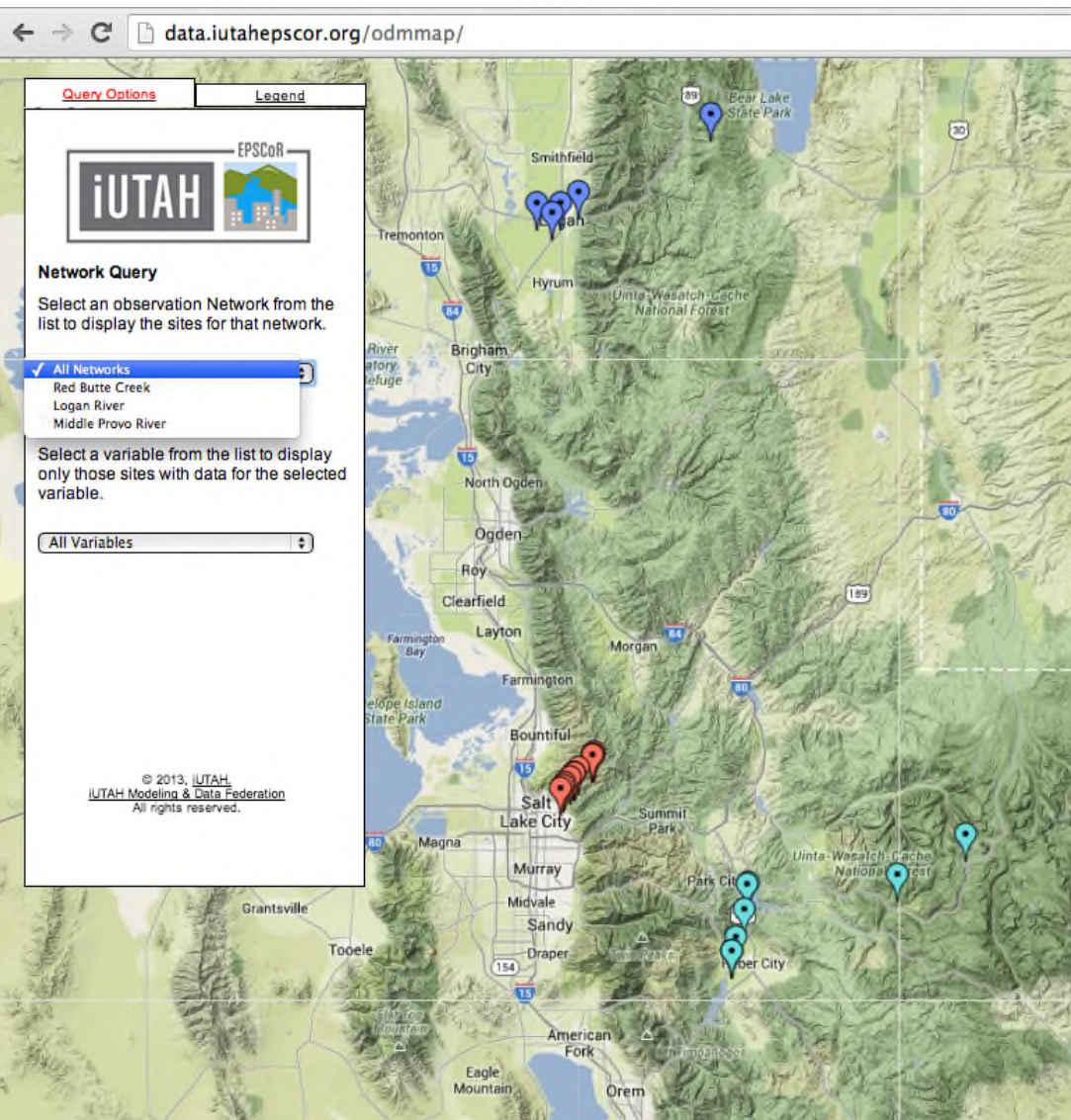
Amber Spackman-Jones



Motivation

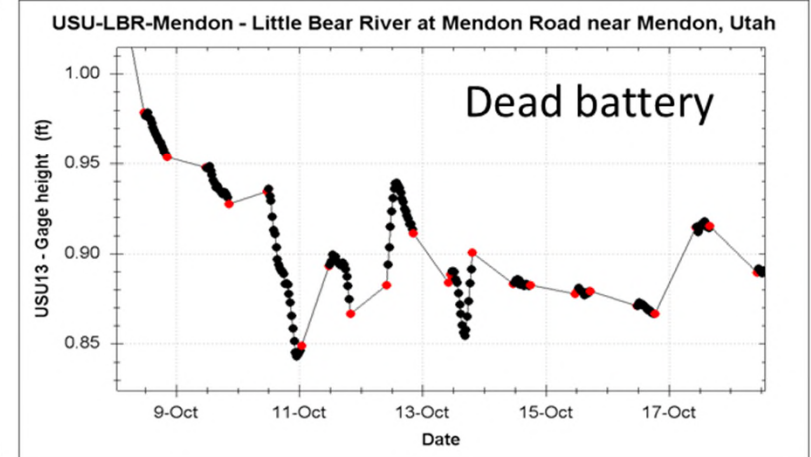
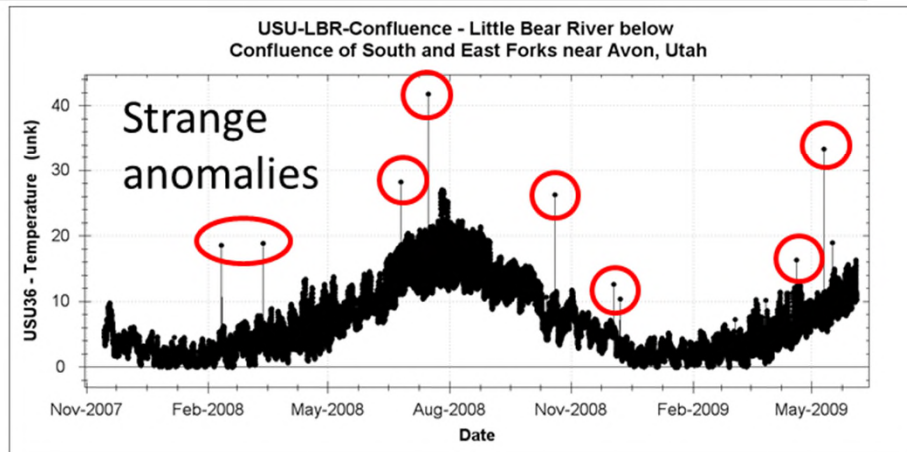
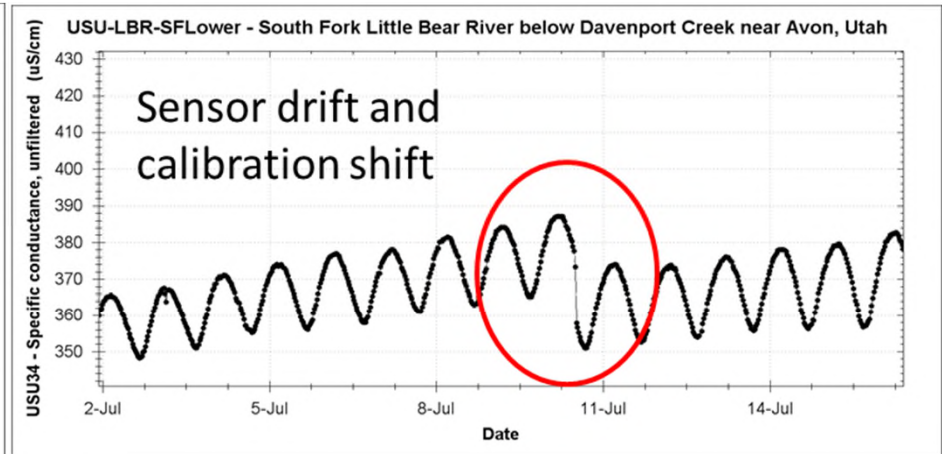
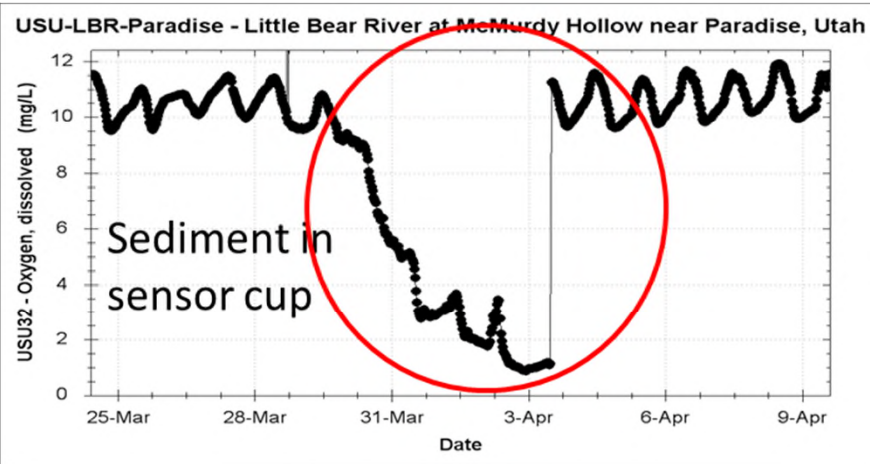


Gradients Along Mountain To Urban Transitions (GAMUT) Network



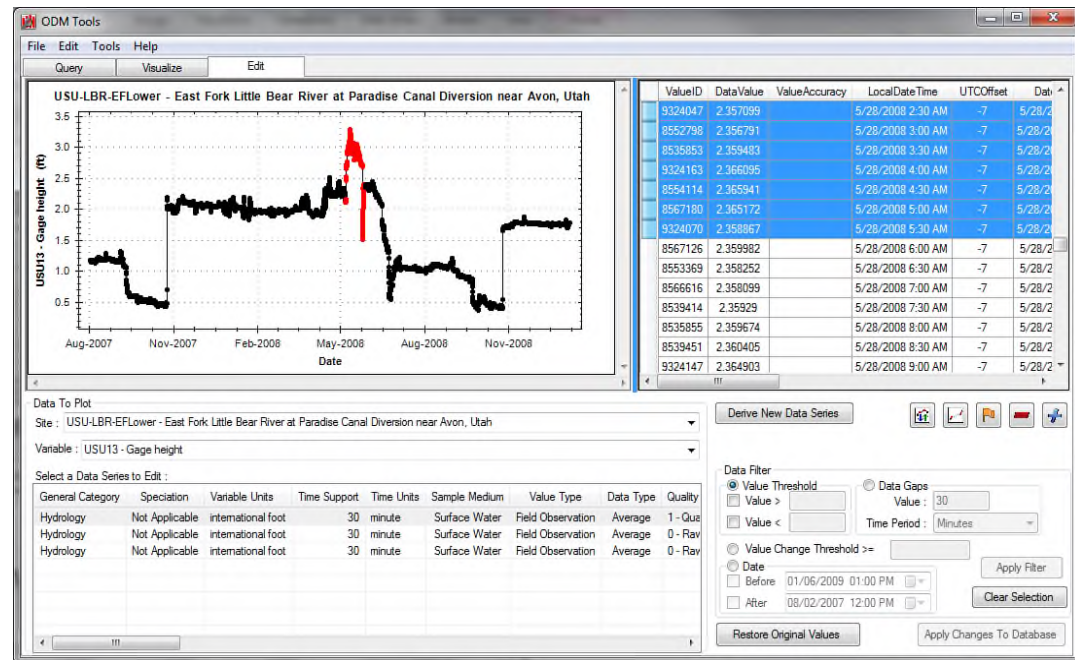
- Ecohydrologic observatory deployed in 3 watersheds: Logan River, Red Butte Creek, Provo River
- Watersheds with similar water source (high elevation snow) but different land use transitions
- Measures aspects of water inputs and outputs and water quality over mountain-to-urban gradient
- Mix of aquatic and terrestrial *in situ* and re-locatable sensors

Motivation

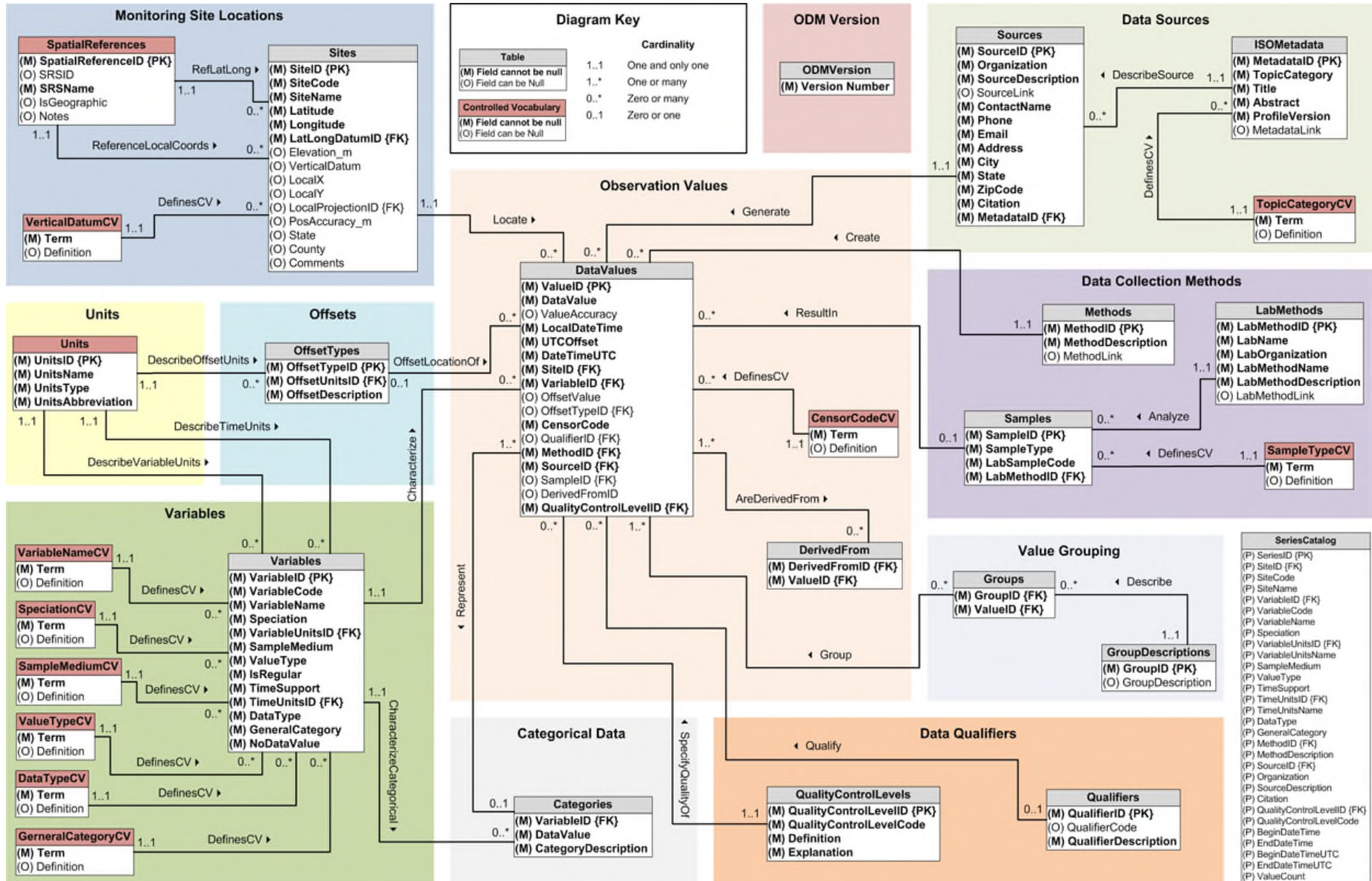


Some History

- ODM Tools originally developed as part of the CUAHSI Hydrologic Information System
- Developed in Microsoft Visual Studio .Net
- Limited to Windows Machines
- Only worked with Microsoft SQL Server databases
- Provided editing tools, but did not preserve the history of edits



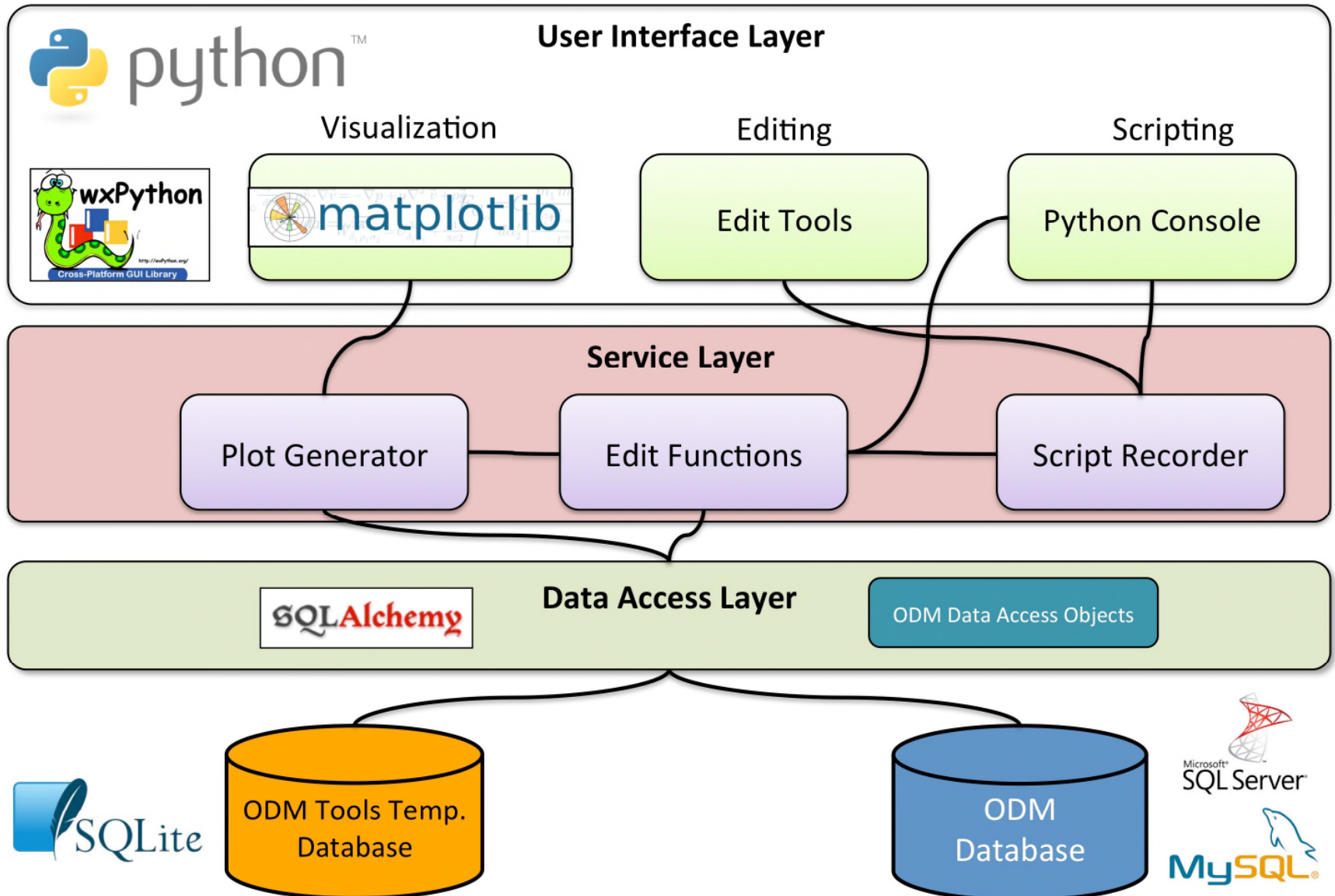
Observations Data Model



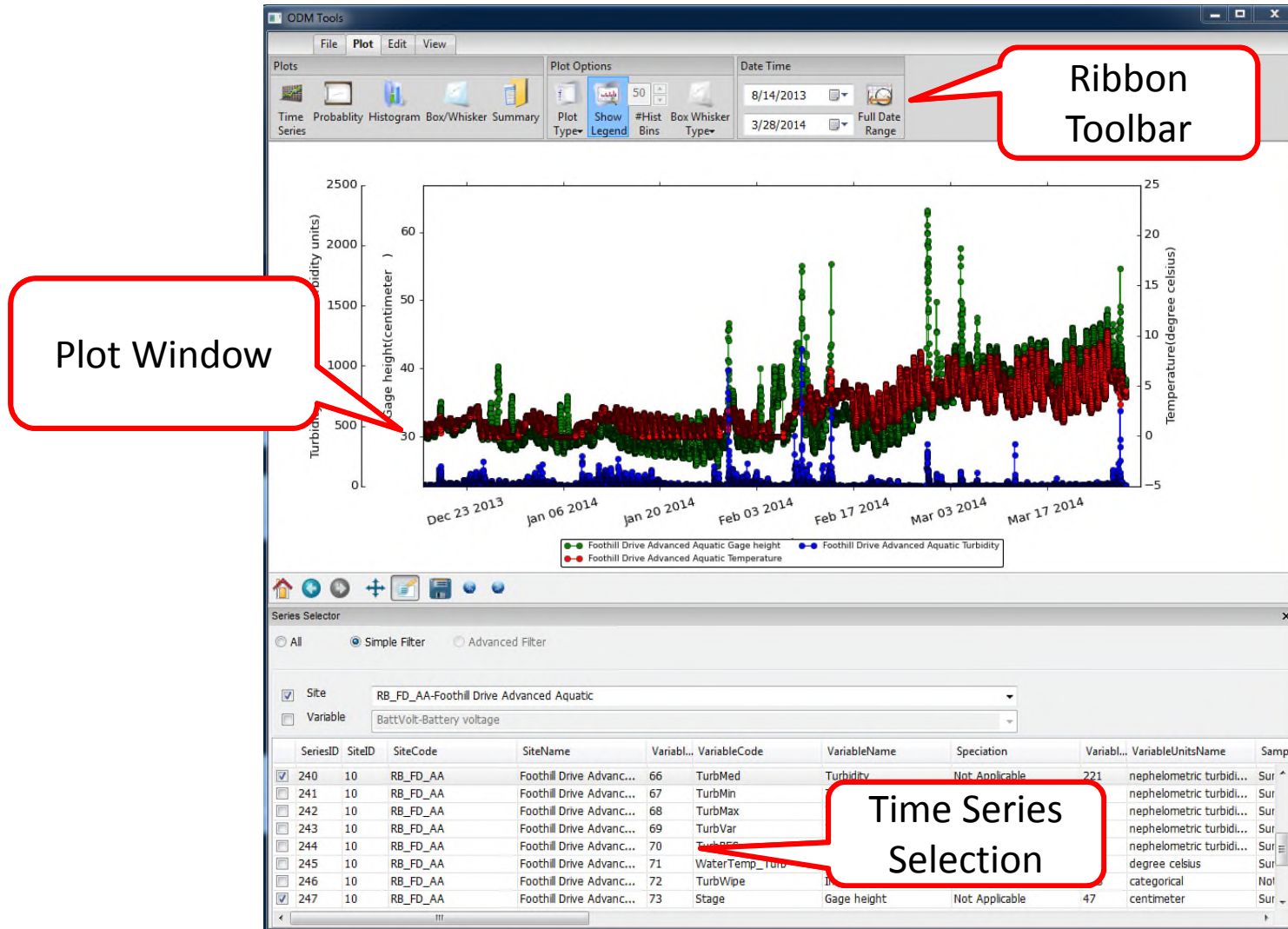
ODM Tools Python: Design Goals

- Multi-platform support (Windows, Linux, Mac)
- Multi-database support (Microsoft SQL Server and MySQL)
- Implement a scripting interface to save the provenance of data edits in QC process
- Modernize the Graphical User Interface (GUI)

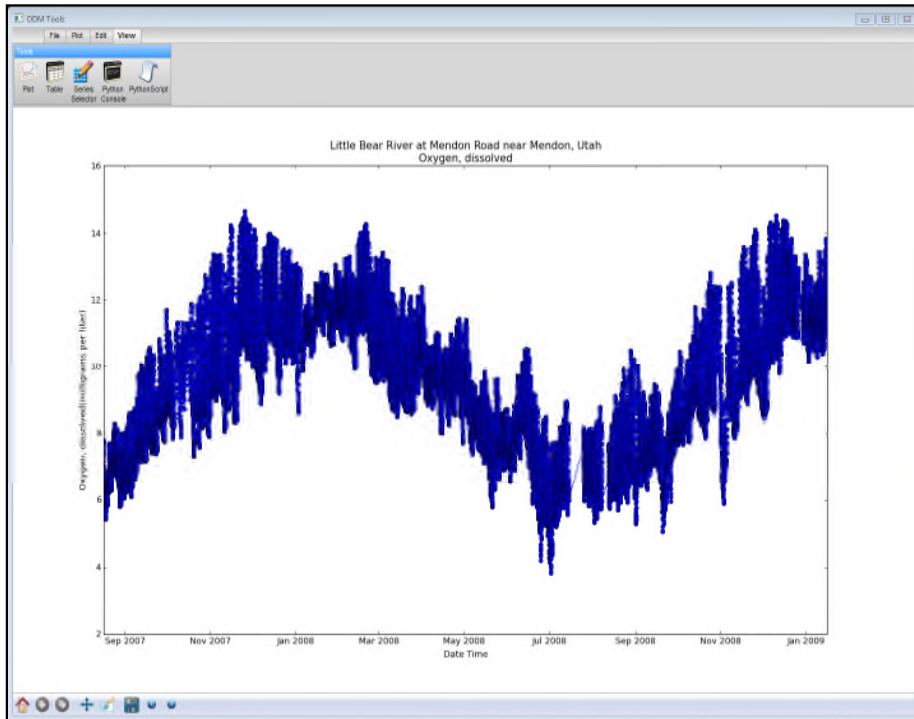
Architecture



Graphical User Interface



Dockable Windows



| ValueID | DataValue | Val... | LocalDateTime | UT... | DateTimeUTC | Sit... | Va... | Of... | O ^ |
|---------|---------------|--------|---------------------|-------|---------------------|--------|-------|-------|-----|
| 8607706 | 4.00045970149 | | 2008-07-02 07:30:00 | -7.0 | 2008-07-02 14:30:00 | 1 | 32 | | |
| 8615287 | 4.25681459701 | | 2008-07-01 07:30:00 | -7.0 | 2008-07-01 14:30:00 | 1 | 32 | | |
| 8616334 | 4.23571177612 | | 2008-07-02 05:30:00 | -7.0 | 2008-07-02 12:30:00 | 1 | 32 | | |
| 8619129 | 4.33288192537 | | 2008-07-02 01:30:00 | -7.0 | 2008-07-02 08:30:00 | 1 | 32 | | |
| 8619782 | 4.4006091194 | | 2008-07-02 03:00:00 | -7.0 | 2008-07-02 10:00:00 | 1 | 32 | | |
| 8619818 | 4.4100449403 | | 2008-07-01 05:00:00 | -7.0 | 2008-07-01 12:00:00 | 1 | 32 | | |
| 8620012 | 4.36762985075 | | 2008-07-02 03:30:00 | -7.0 | 2008-07-02 10:30:00 | 1 | 32 | | |
| 8620220 | 3.83081343284 | | 2008-07-02 08:00:00 | -7.0 | 2008-07-02 15:00:00 | 1 | 32 | | |
| 8622377 | 3.89308523881 | | 2008-07-02 06:30:00 | -7.0 | 2008-07-02 13:30:00 | 1 | 32 | | |
| 8622441 | 4.49635820896 | | 2008-07-01 04:30:00 | -7.0 | 2008-07-01 11:30:00 | 1 | 32 | | |
| 8625349 | 4.58550746269 | | 2008-07-02 00:30:00 | -7.0 | 2008-07-02 07:30:00 | 1 | 32 | | |
| 8625395 | 4.59515373134 | | 2008-07-02 00:00:00 | -7.0 | 2008-07-02 07:00:00 | 1 | 32 | | |
| 8625705 | 4.2038311791 | | 2008-06-25 06:30:00 | -7.0 | 2008-06-25 13:30:00 | 1 | 32 | | |
| 8626475 | 4.62864941791 | | 2008-07-03 04:30:00 | -7.0 | 2008-07-03 11:30:00 | 1 | 32 | | |
| 8627214 | 4.66996268657 | | 2008-07-03 04:00:00 | -7.0 | 2008-07-03 11:00:00 | 1 | 32 | | |
| 8627341 | 4.66188076119 | | 2008-07-03 02:00:00 | -7.0 | 2008-07-03 09:00:00 | 1 | 32 | | |
| 8628198 | 4.28373167164 | | 2008-07-01 05:30:00 | -7.0 | 2008-07-01 12:30:00 | 1 | 32 | | |
| 8630093 | 4.78889435821 | | 2008-07-02 10:00:00 | -7.0 | 2008-07-02 17:00:00 | 1 | 32 | | |
| 8630222 | 4.37667031343 | | 2008-07-02 04:30:00 | -7.0 | 2008-07-02 11:30:00 | 1 | 32 | | |
| 8634141 | 4.93983449254 | | 2008-06-30 07:30:00 | -7.0 | 2008-06-30 14:30:00 | 1 | 32 | | |
| 8634276 | 4.48706534328 | | 2008-06-25 05:00:00 | -7.0 | 2008-06-25 12:00:00 | 1 | 32 | | |

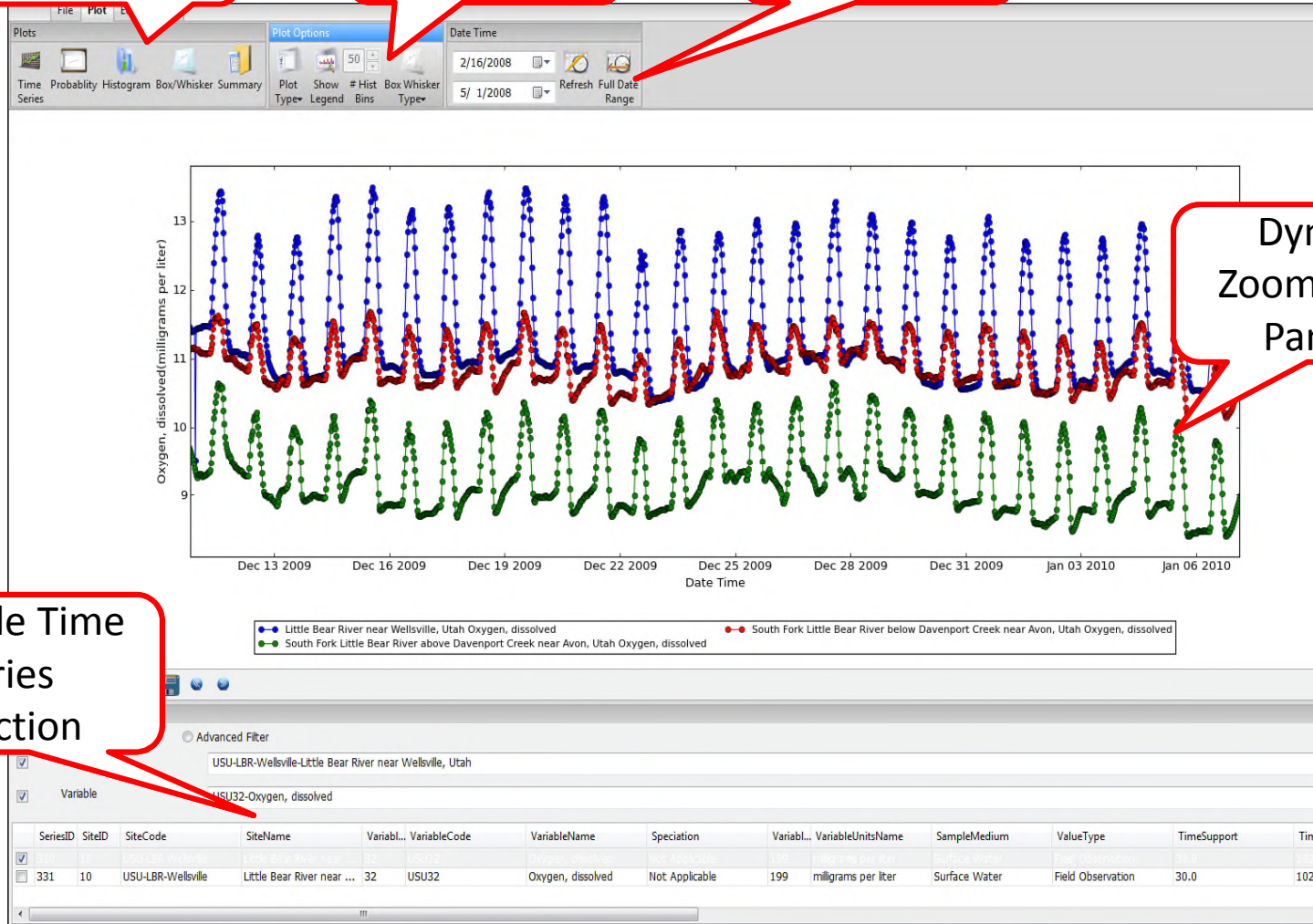
| Series Selector | | | | | | | | |
|-------------------------------------|--------|----------|----------------|----------------------------|--------------|--------------|--------------------------|-------------------|
| All Simple Filter Advanced Filter | | | | | | | | |
| SeriesID | SiteID | SiteCode | SiteName | Variabl... | VariableCode | VariableName | Speciation | Varia... |
| <input type="checkbox"/> | 12 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 13 | USU13 | Gage height | Not Applicable 48 |
| <input type="checkbox"/> | 13 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 13 | USU13 | Gage height | Not Applicable 48 |
| <input type="checkbox"/> | 14 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 13 | USU13 | Gage height | Not Applicable 48 |
| <input checked="" type="checkbox"/> | 15 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 32 | USU32 | Oxygen, dissolved | Not Applicable 19 |
| <input type="checkbox"/> | 16 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 32 | USU32 | Oxygen, dissolved | Not Applicable 19 |
| <input type="checkbox"/> | 17 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 33 | USU33 | Oxygen, dissolved per... | Not Applicable 1 |
| <input type="checkbox"/> | 18 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 34 | USU34 | Specific conductance | Not Applicable 19 |

Data Visualization

Multiple
Plot Types

Plot Display
Options

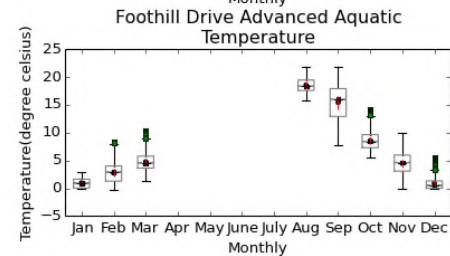
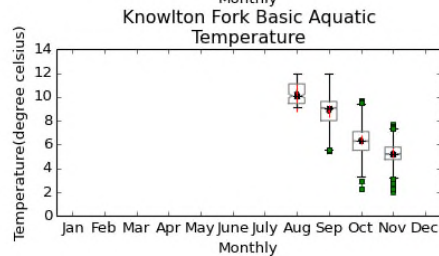
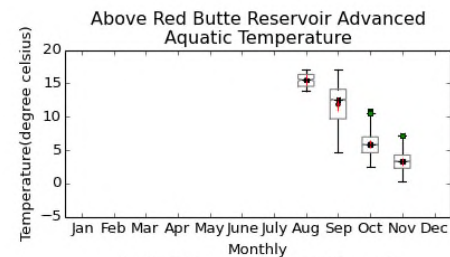
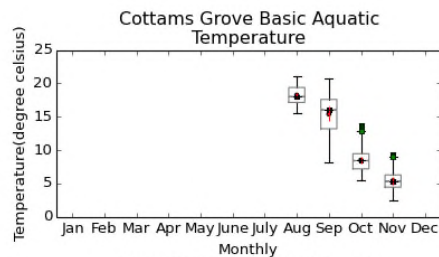
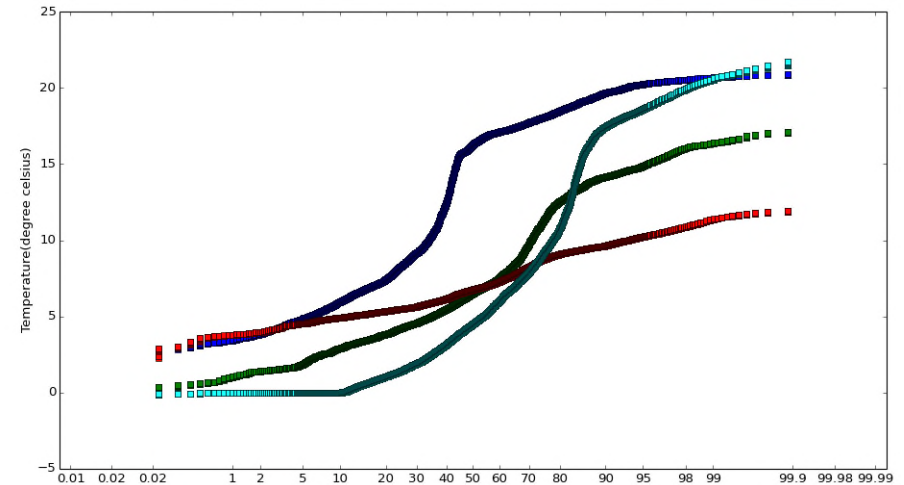
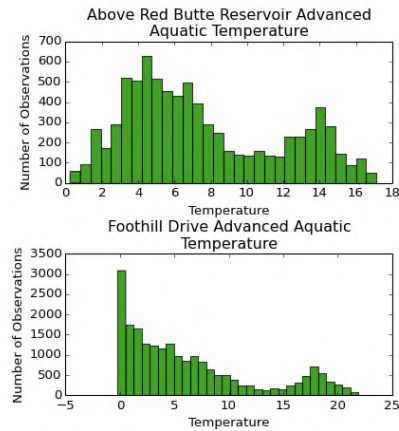
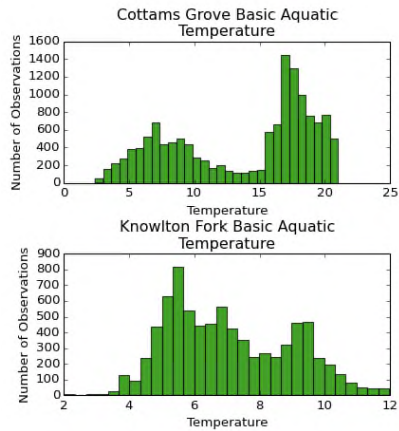
Date Range
Restrictions



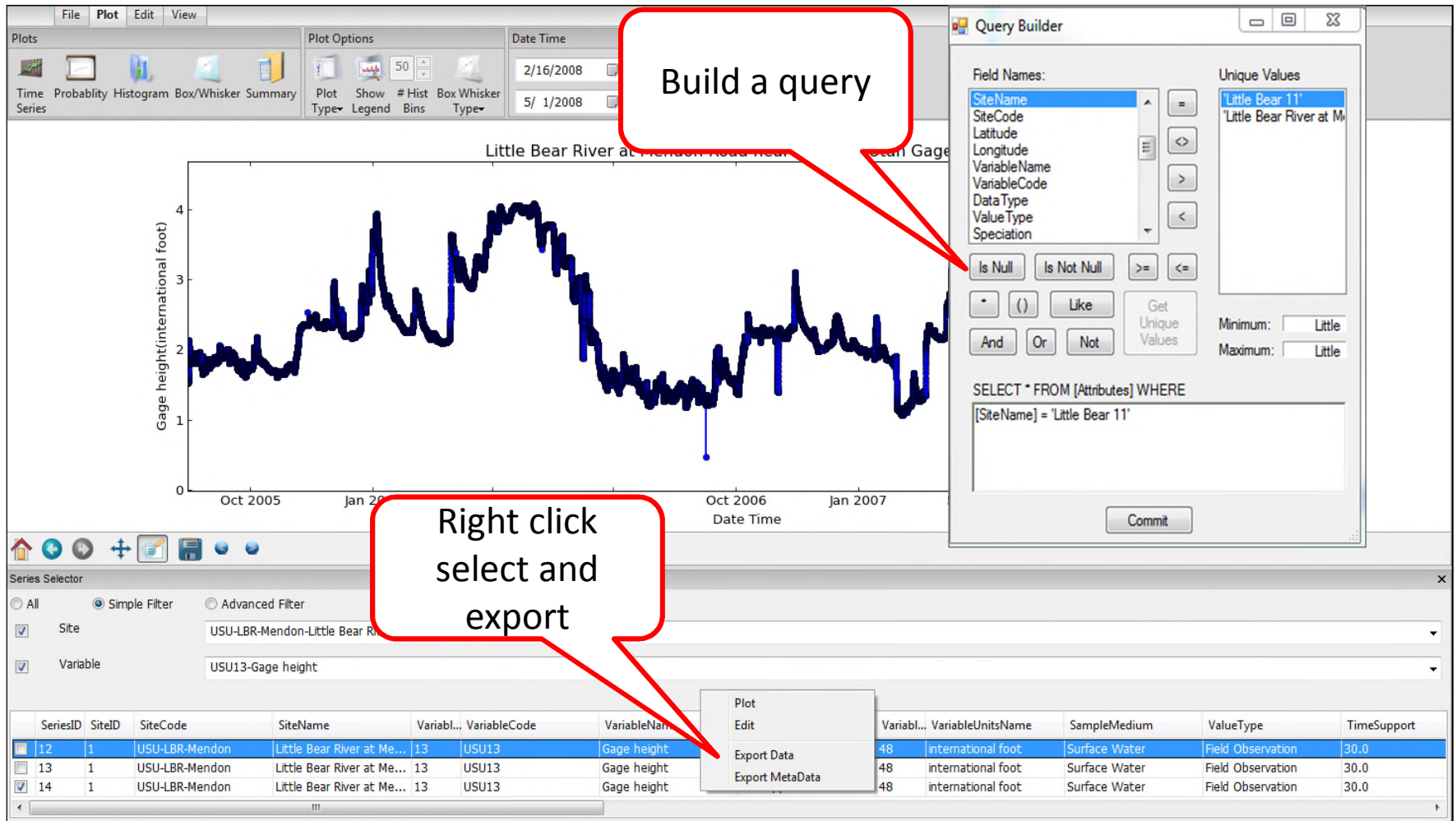
Multiple Time
Series
Selection

Dynamic
Zooming and
Panning

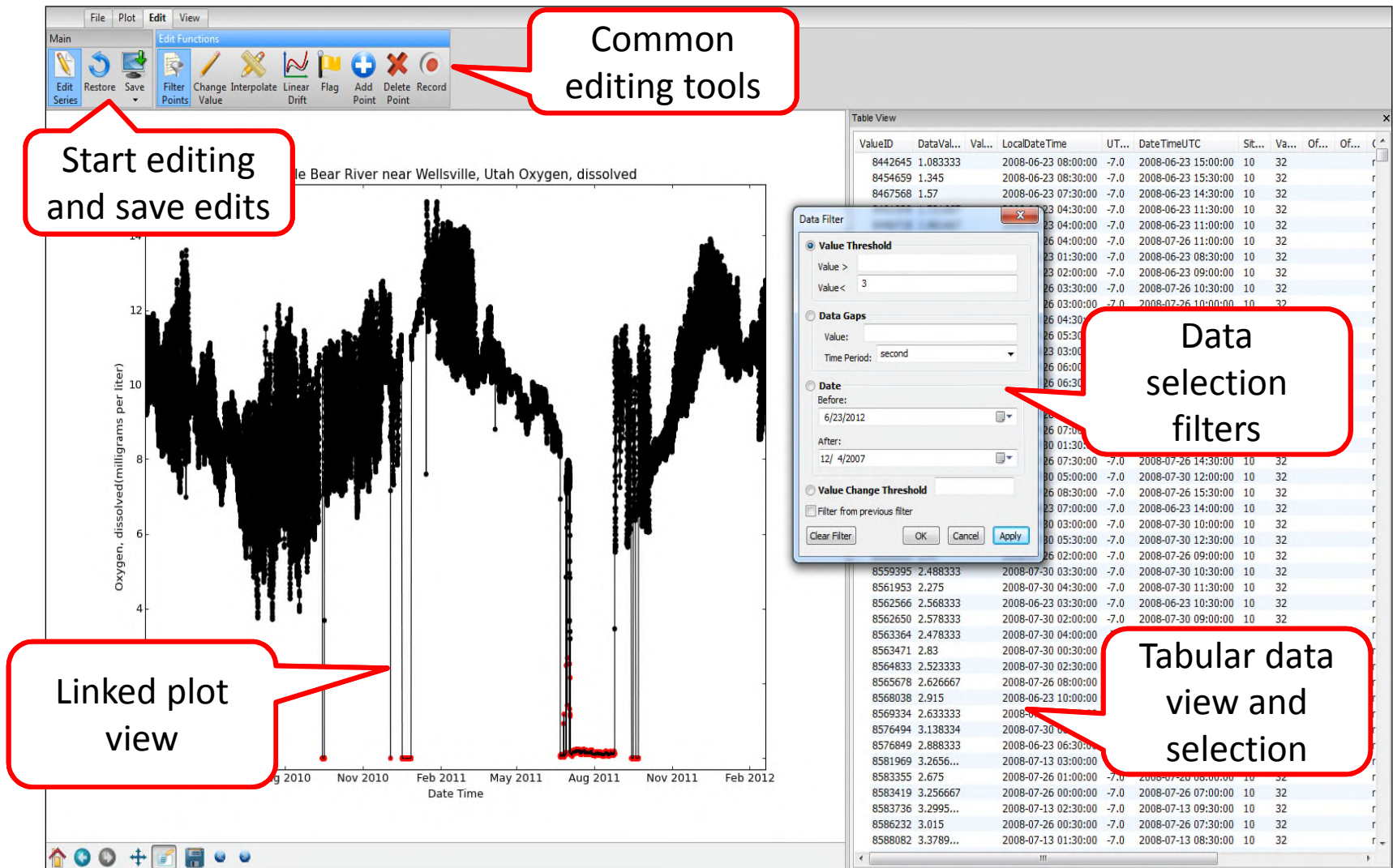
Data Visualization: Plot Types



Query and Export



Data Editing for Quality Control



Data Editing for Quality Control

The screenshot displays the ODM Tools application interface. The main window shows a plot titled "Little Bear River at Paradise, Utah Gage height" with the y-axis labeled "Gage height(international foot)" ranging from 4 to 10 and the x-axis labeled "Date Time" from Dec 22 2007 to Apr 12 2008. The plot shows a significant jump in data around March 2008, with a red line indicating the edited data. Below the plot is a "Series Selector" table. To the right, a "Python Console" window shows the execution of Python code to edit the data.

Series Selector Table:

| SeriesID | SiteID | SiteCode | SiteName | VariableCode | VariableName | Speciation | VariableUnitsName | SampleM |
|----------|--------|--------------------|----------------------------|--------------|--------------------------|----------------|--------------------------|---------|
| 165 | 11 | USU-LBR-Confluence | Little Bear River below... | USU36 | Temperature | Not Applicable | 96 degree celcius | Surfa |
| 166 | 11 | USU-LBR-Confluence | Little Bear River below... | USU39 | Phosphorus, total as P | | 199 milligrams per liter | Surfa |
| 167 | 11 | USU-LBR-Confluence | Little Bear River below... | USU40 | Phosphorus, total as ... | P | 199 milligrams per liter | Surfa |
| 168 | 11 | USU-LBR-Confluence | Little Bear River below... | USU41 | Solids, total Suspended | Not Applicable | 199 milligrams per liter | Surfa |
| 169 | 12 | 10105900 | Little Bear River at Pa... | USU42 | Gage height | Not Applicable | 48 international foot | Surfa |
| 170 | 12 | 10105900 | Little Bear River at Pa... | USU43 | Discharge | Not Applicable | 35 cubic feet per second | Surfa |
| 171 | 12 | 10105900 | Little Bear River at Pa... | USU42 | Gage height | Not Applicable | 48 international foot | Surfa |

Python Console Code:

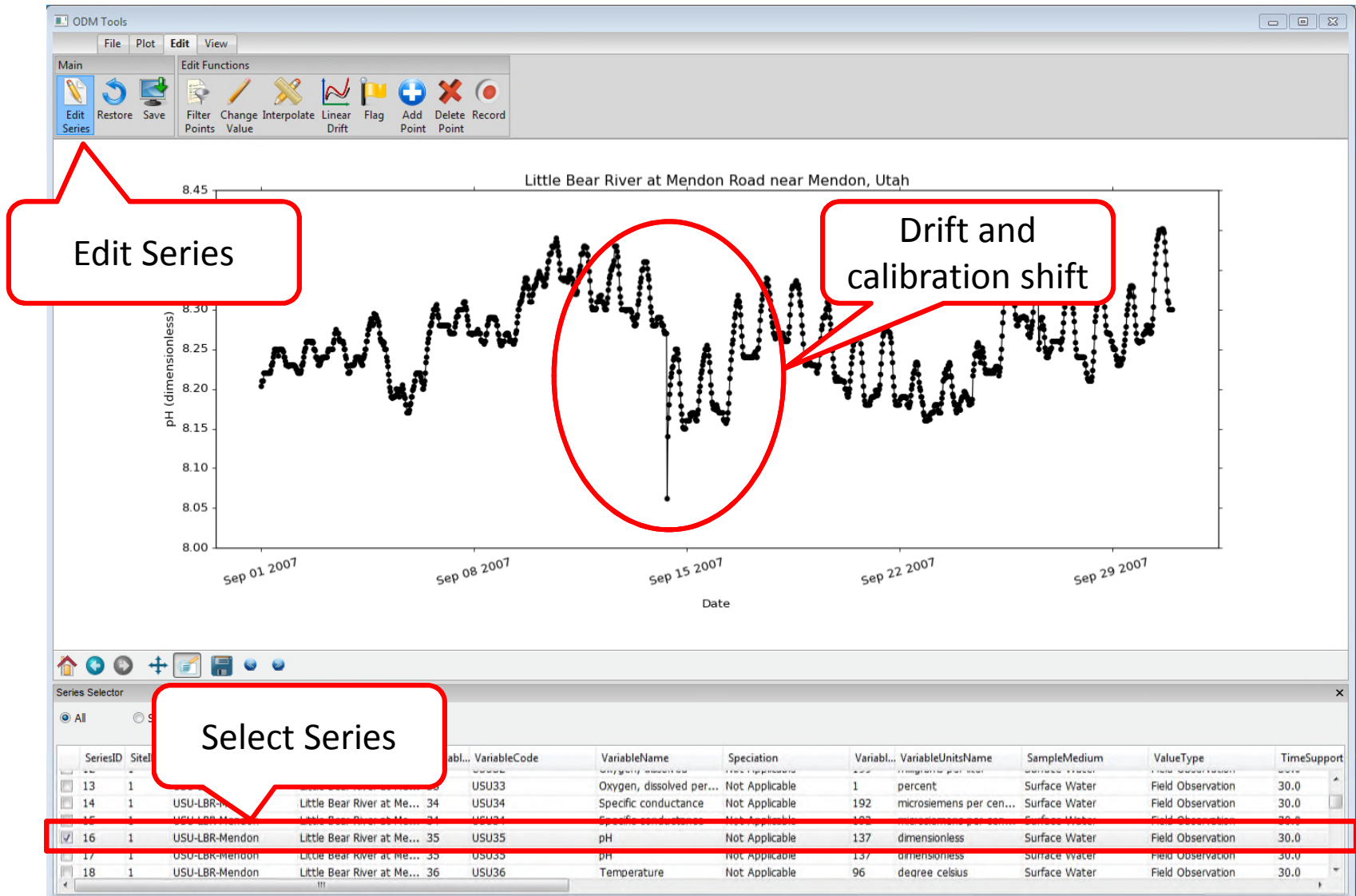
```
>>>
>>> from odmservices.edit_service import EditService
>>> series = EditService(series_id= 171,
... connection_string =
... "mssql+pyodbc://ODM:odm@localhost/LittleBear11")
>>> series.filter_value(8, '>')
>>>
>>> series.get_active_points()
[(2766974, 9.04, datetime.datetime(2008, 3, 2, 11, 30)), (
2766975, 9.04, datetime.datetime(2008, 3, 2, 11, 45)), (2766976
, 9.04, datetime.datetime(2008, 3, 2, 12, 0)), (2766977, 9.04,
datetime.datetime(2008, 3, 2, 12, 15)), (2766978, 9.04,
datetime.datetime(2008, 3, 2, 12, 30)), (2766979, 9.04,
datetime.datetime(2008, 3, 2, 12, 45)), (2766980, 9.04,
datetime.datetime(2008, 3, 2, 13, 0)), (2766981, 9.04, datetime
.datetime(2008, 3, 2, 13, 15)), (2766982, 9.05, datetime.
datetime(2008, 3, 2, 13, 30)), (2766983, 9.06, datetime.
datetime(2008, 3, 2, 13, 45)), (2766984, 9.06, datetime.
datetime(2008, 3, 2, 14, 0)), (2766985, 9.08, datetime.datetime
(2008, 3, 2, 14, 15)), (2766986, 9.09, datetime.datetime(2008,
3, 2, 14, 30)), (2766987, 9.1, datetime.datetime(2008, 3, 2, 14
, 45)), (2766988, 9.1, datetime.datetime(2008, 3, 2, 15, 0)), (
2766989, 9.1, datetime.datetime(2008, 3, 2, 15, 15)), (2766990,
9.1, datetime.datetime(2008, 3, 2, 15, 30)), (2766991, 9.1,
datetime.datetime(2008, 3, 2, 15, 45)), (2766992, 9.09,
```

Python Script Editor:

```
1 from odmservices.edit_service import EditService
2 series = EditService(series_id= 171,
3 connection_string = "mssql+pyodbc://ODM:odm@localhost/Lit
4 series.filter_value(8, '>')
5
```

How does it work?

Step 1: Select a Time Series for Editing



Step 2: Select Data to Edit

The screenshot displays the ODM Tools software interface. The main window shows a plot titled "Little Bear River at Mendon Road near Mendon, Utah" with pH (dimensionless) on the y-axis (ranging from 8.00 to 8.45) and time on the x-axis (from Sep 01 2007 to Sep 22 2007). The plot shows two data series: one in red (pH values between 8.15 and 8.40) and one in black (pH values between 8.05 and 8.35). A red circle highlights the "Edit Functions" menu, and a red arrow points from it to the "Data Filter" dialog box.

The "Data Filter" dialog box is open, showing the following options:

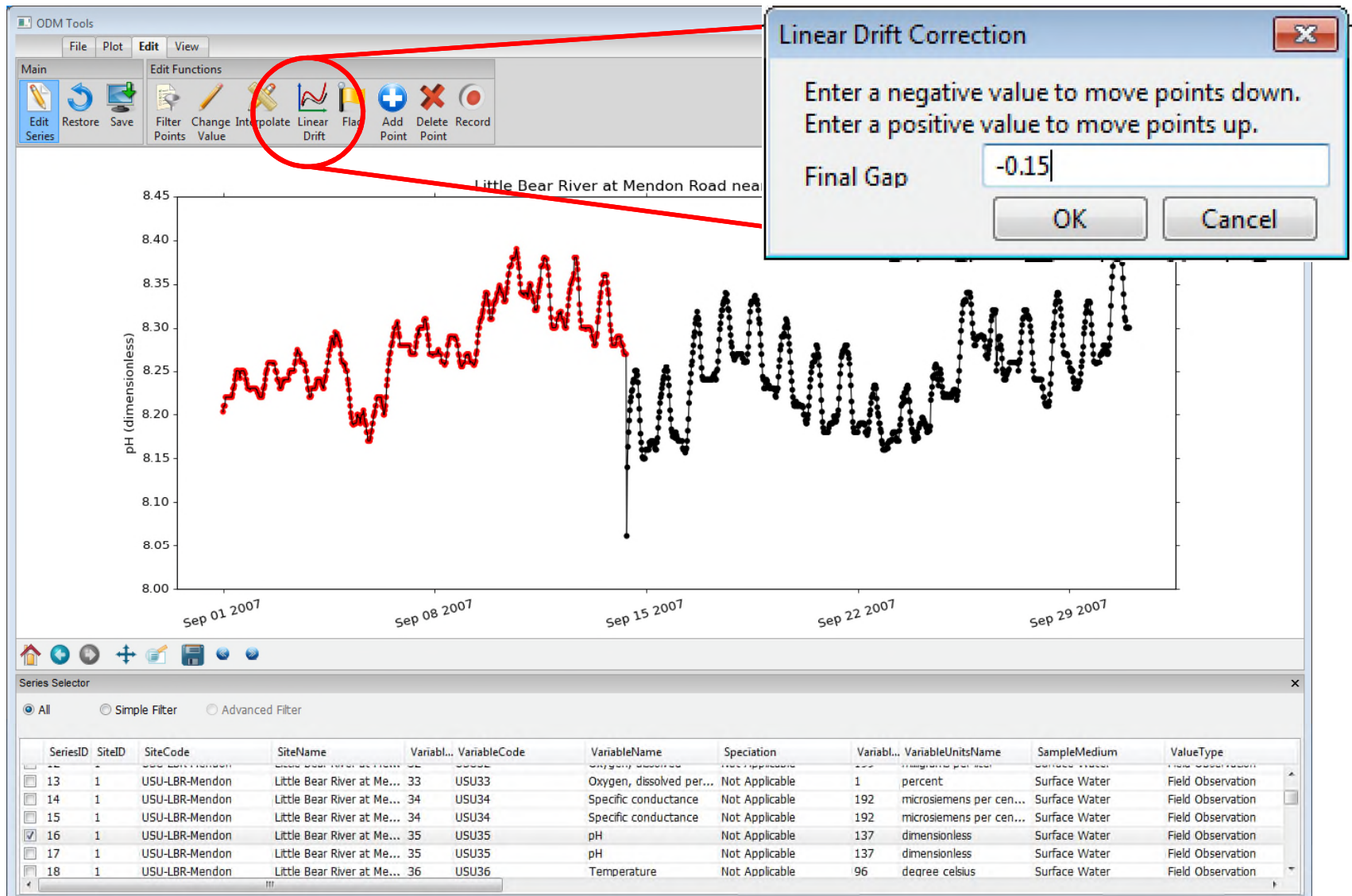
- ☒ **Value Threshold**
 - Value >
 - Value <
- ☐ **Data Gaps**
 - Value:
 - Time Period:
- ☐ **Date**
 - Before:
 - After:
- ☐ **Value Change Threshold**
- ☐ Filter from previous filter

Buttons at the bottom: Clear Filter, OK, Cancel, Apply.

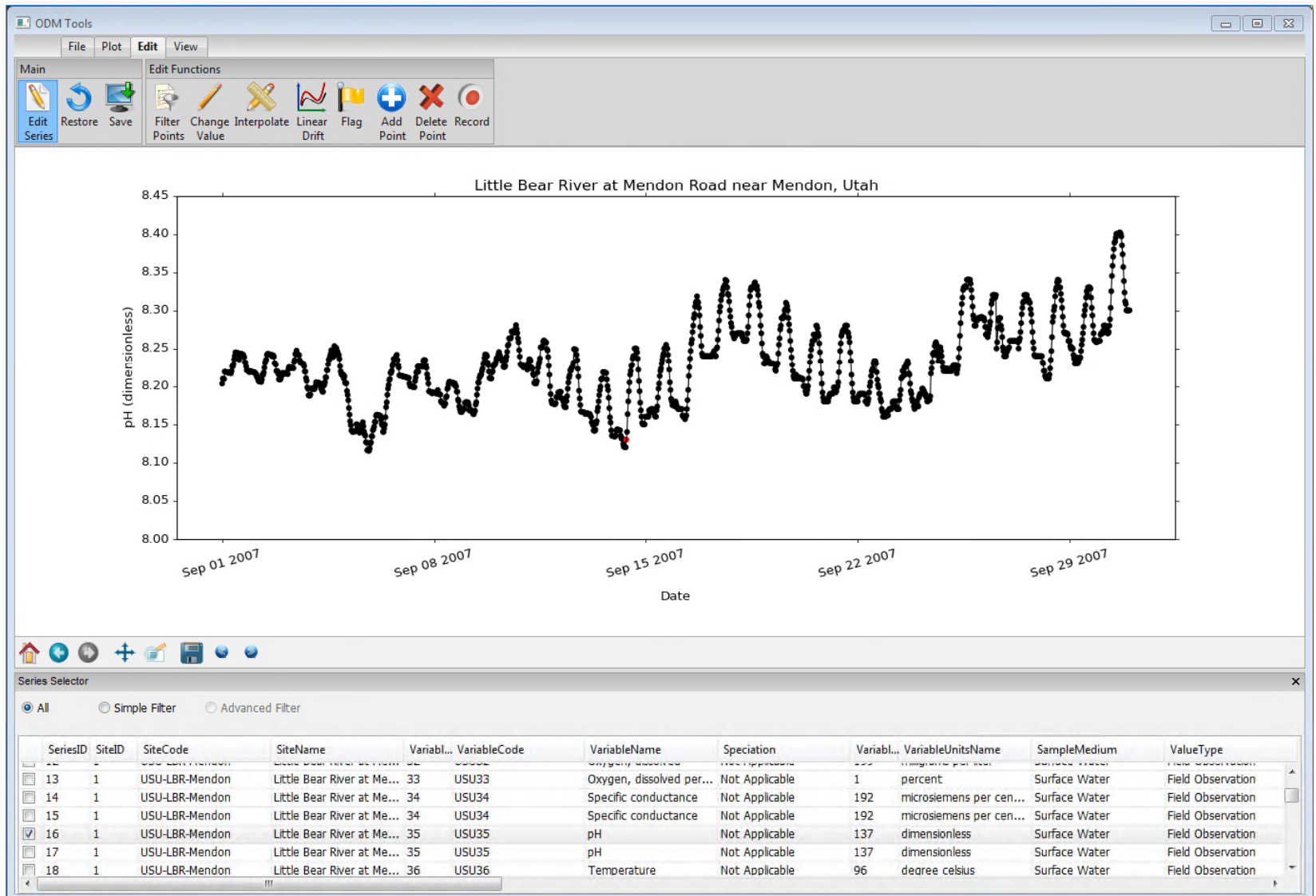
The "Series Selector" at the bottom shows a list of data series. The selected series is "pH" (SeriesID 16, SiteID 1, VariableCode 137).

| SeriesID | SiteID | SiteCode | SiteName | VariableCode | VariableName | Speciation | VariableCode | VariableName | Speciation | VariableCode | VariableName | Speciation |
|----------|--------|----------------|----------------------------|--------------|--------------|--------------------------|----------------|--------------|-------------------------|---------------|-------------------|------------|
| 13 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 33 | USU33 | Oxygen, dissolved per... | Not Applicable | 1 | percent | Surface Water | Field Observation | |
| 14 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 34 | USU34 | Specific conductance | Not Applicable | 192 | microsiemens per cen... | Surface Water | Field Observation | |
| 15 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 34 | USU34 | Specific conductance | Not Applicable | 192 | microsiemens per cen... | Surface Water | Field Observation | |
| 16 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 35 | USU35 | pH | Not Applicable | 137 | dimensionless | Surface Water | Field Observation | |
| 17 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 35 | USU35 | pH | Not Applicable | 137 | dimensionless | Surface Water | Field Observation | |
| 18 | 1 | USU-LBR-Mendon | Little Bear River at Me... | 36 | USU36 | Temperature | Not Applicable | 96 | degree celsius | Surface Water | Field Observation | |

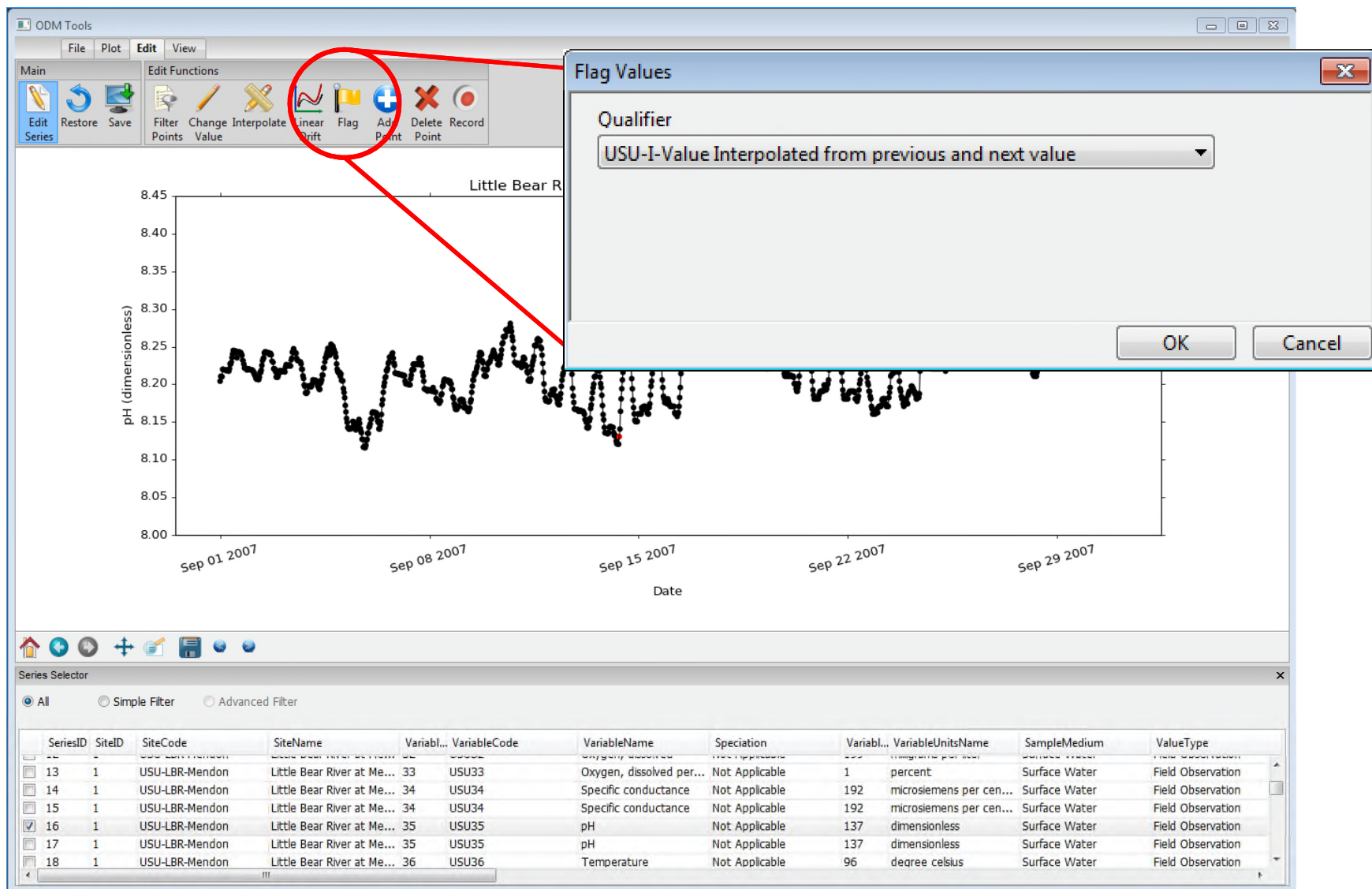
Step 3: Linear Drift Correction



Step 4: Interpolate



Step 5: Flag



Step 6: Save Modified Data Series

Save...

How would you like to save this data series?

☐ Save as new data series

☒ Save as modified data series

Save...

☒ Automatically generate a new method

☐ Select an existing method

Description

Water temperature measured in the river

Turbidity measured in the river

Water level measured in the river

Battery voltage measured in the river

Air temperature measured in the river

Relative humidity measured in the river

Precipitation measured in the river

☐ Create a new Method

Method Description

Save...

☒ Select an existing method

| Code | Definition |
|-------|-------------------|
| -9999 | Unknown |
| 0 | Raw data |
| 1 | Quality control |
| 2 | Derived data |
| 3 | Intermediate data |
| 4 | Known data |

☐ Create Quality Control

Code:

Definition:

Explanation:

Save...

☒ Use Current Variable

☐ Select an existing variable

| Code | Name |
|------|-----------------|
| USU3 | Battery voltage |
| USU4 | Turbidity |
| USU5 | Turbidity |
| USU6 | Turbidity |
| USU7 | Turbidity |
| USU8 | Turbidity |

☐ Create New Variable

Save...

Summary

Site

Code: USU-LBR-Mendon

Name: Little Bear River at Mendon Road near Mendon, Utah

Variable

Code: USU35

Name: pH

Units: dimensionless

Sample Medium: Surface Water

Value Type: Field Observation

Time Support: 30.0

Time Units: minute

Data Type: Average

General Category: Water Quality

Method

Description: Values derived from ODM Tools Python

Source

Organization: Utah State University Utah Water Research Laboratory

Description: Continuous water quality monitoring by Utah State University

Citation: Continuous water quality monitoring by Jeff Horshburgh, David Ste...

< Back

Finish

Cancel

Recording Edits

The screenshot displays the ODM Tools interface. The main window shows a plot titled "Little Bear River at Mendon Road near Mendon, Utah" with pH (dimensionless) on the y-axis (ranging from 8.00 to 8.45) and Date on the x-axis (ranging from Aug 30 2007 to Sep 27 2007). The plot shows a fluctuating line with data points. A red circle highlights the "Record" button in the "Edit Functions" toolbar.

Below the plot is the "Series Selector" window, which lists various data series. The series are organized by SiteID, SiteCode, SiteName, VariableCode, VariableName, and Speciation. The series "Little Bear River at Me... 35 USU35 pH" is selected.

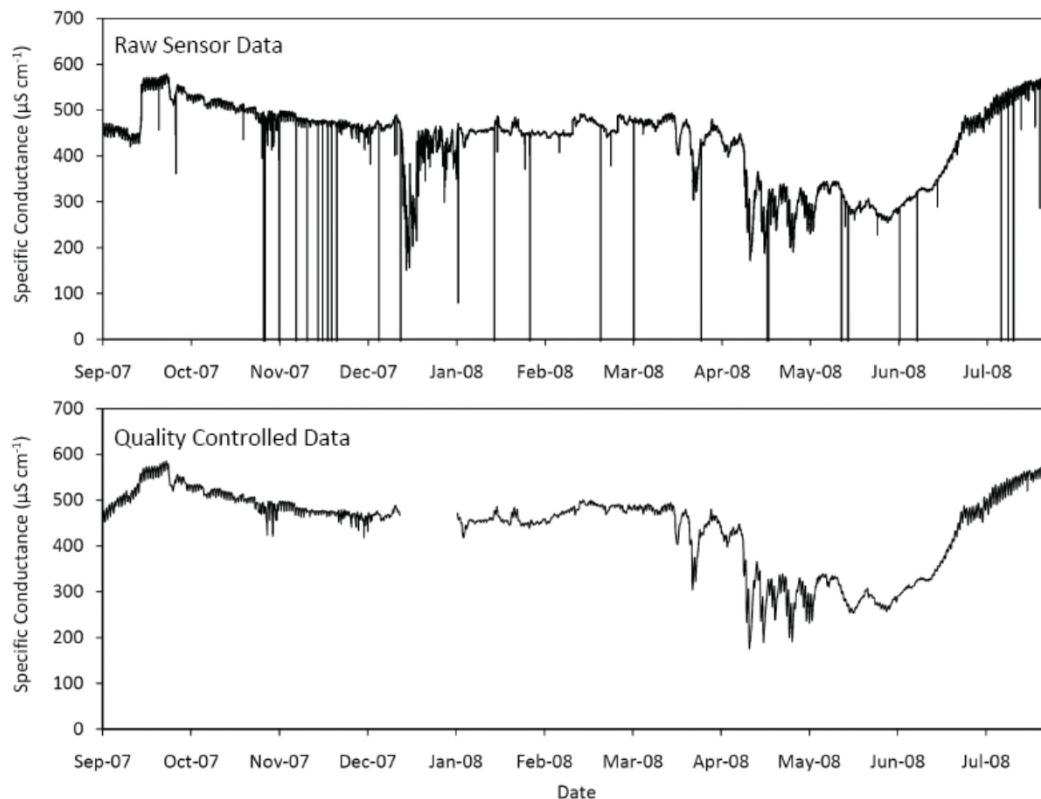
On the right side, a window titled "Editing a new file" shows a Python code editor with the following code:

```
1 from odmservices import EditService
2 from odmservices import SeriesService
3 edit_service = EditService(series_id=16, connection_string='mssql+pyodbc://sa:nlcd34GI:
4 series_service = SeriesService(connection_string='mssql+pyodbc://sa:nlcd34GI:
5 ## To run commands from the python console uncomment and run the following c
6 #edit_service = Tools
7 edit_service.select_points([datetime.datetime(2007, 9, 1, 0, 0), datetime.da
8 edit_service.drift_correction(-0.15)
9 edit_service.select_points([datetime.datetime(2007, 9, 14, 8, 30)])
10 edit_service.interpolate()
11
```

A red speech bubble points to the code editor with the text: "Automatically generated Python code with each editing step".

Ultimate Objective

- Transform raw sensor data to quality controlled data in a repeatable way



Summary

- ODM Tools Python is a cross platform (Windows, Mac) software for sensor data management
- Visualization capabilities are helpful in screening new data as they arrive
- ODM Tools provides GUI-based and scripting of data quality control edits

This project is funded by National
Science Foundation grant EPS-1208732.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.



Questions?

OPEN SOURCE CODE REPOSITORY:

ODM Tools Python is available in GitHub

<https://github.com/UCHIC/ODMToolsPython>

